UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/672,368	09/28/2000	Francis X. McKeen	042390.P9575	7652

7590 09/06/2006

Blakely Sokoloff Taylor & Zafman LLP
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025

EXAMINER	
HO, THOMAS M	
ART UNIT	**PAPER NUMBER**
2132	

DATE MAILED: 09/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/672,368	**Applicant(s)** MCKEEN ET AL.	
	Examiner Thomas M. Ho	**Art Unit** 2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 June 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. The amendment of 6/21/06 has been received and entered.

2. Claims 1-15 are pending.

Response to Arguments

The Applicant has argued:

Memory is typically not attached to an interface bus like that shown in Summers (instead, it is usually accessed through a memory controller that coordinates interactions from the processors(s) and peripherals(s). Summers does not teach or suggest interposing his isolated bus tranceivers between a memory and a data bus, nor would such interposition be likely to work. Summers is directed at isolated insecure or untrusted peripheral devices temporarily when sensitive data is being placed on a date bus. System memory is used by processors and all peripheral, trusted or untrusted, and the mix of transactions that involve the memory.

Applicant appears to be advocating a very specific technical implementation of the embodiment and has chosen to read a somewhat arbitrary arrangement of an implementation into the 35 USC 103 combination rendered, and argued why such an implementation would not be feasible.

The Examiner contends however that the rejection only broadly called for the combination of a the technology of the secure bus arbiter taught by Summers with Coulouris et al. and Silberschatz et al.

The rejection under 35 USC 103 is not a blueprint which dictates the exact technical specifications by which two references may be combined. The rejection under 35 USC 103 seeks to combine the general technology of the secure bus arbiter taught by Summers into yet another hypothetical combination between Coulouris et al. and Silberschatz et al. To this effect, reasons for the advantages of such a combination have been given.

Applicant's argument is analogous to the hypothetical situation given below:

Suppose the Applicant claimed a station wagon with red doors.

Suppose, the Examiner finds two references:

- Reference A teaches a station wagon.
- Reference B teaches a minivan with red doors and that placing red doors on a car provides a substantial decrease in aerodynamic drag causing an increase in fuel efficiency.

Suppose then that as a result, the Examiner combines reference A and reference B in a 35 USC 103 rejection.

It can certainly be argued that a minivan engine would not be compatible into a station wagon without substantial modification to the minivan engine.

It can be argued that a station wagon exhaust system would not typically support the output of the minivan.

It can be argued that a minivan door would never fit on a station wagon, and therefore such a combination would be infeasible. Even if the Examiner found a reference B which taught a station wagon with red doors, the Applicant could argue that the red doors from one model of station wagon would never "just fit" the door construct of another model of station wagon.

These arguments derive from vagueness of the technical implications which arise out of virtually any rejection under 35 USC 103. The technical question and details of whether or not the combination would involve the exchanging of exhaust systems or engines, or how the modification of minivan-doors to fit a station wagon chassis does not need to be answered by the rejection.

For this reason, the Examiner believes that the contentions the Applicant has brought up in page 5, last paragraph – page 6 first paragraph arguing that "interposing his isolating bus tranceivers between a memory and data bus is not likely to work" are unconvincing.

The rejection only broadly called for the usage of a secure arbiter bus module, while the Applicant is arguing why his specific technical arrangement of the combination of reference would not be likely to succeed.

Not only does the Applicant not provide any evidence for the allegation that such an imposition would not be likely to work, but the rejection has not confined itself to the specific technical arrangement the Applicant has argued for.

Applicant's additional arguments paragraph appears to be another form of Applicant's previous argument, arguing different yet specific technical arrangements.

These arguments were not found by the Examiner to be persuasive. Accordingly the rejections have been maintained.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coulouris et al. and Silberschatz et al. and Summers et al., US patent 6098133.

In reference to claim 1:

(Coulouris et al. Section 6.3 Processes and Threads) discloses a method comprising:

- Identifying if an event is one of a class of events to be handled in the isolated execution mode, where the isolated execution mode is a processor running a secure process (Page 168), and the event is one of an event or events that might be handled by that process, where threads within a process have their own software interrupt handling mechanisms
- Handling the event using the first page table map if the event is identified as one of the class of events to be handled by the isolated execution mode, where the first page table map is the virtual memory map which maps the memory for the running processes(page 169, 190-192), and the event identified as one of the events to be handled by the isolated execution mode is an event that is to be handled by that process. (page 172)

Coulouris et al. does not explicitly disclose

- Maintaining a first page table map for use in an isolated execution mode and a second page table map for use in a normal execution mode.
- Dynamically swapping between the first page table map and the second page table map responsive to a change in execution mode.

Silberschatz et al. (p 270-271) discloses

- Maintaining a first page table map for use in an isolated execution mode and a second page table map for use in a normal execution mode, where the first page table map is a standard process which executes its own code in an isolated manner, and the normal execution mode is the special case of shared pages between processes.

- Dynamically swapping between the first page table map and the second page table map responsive to a change in execution mode, where processes are isolated execution modes and changing from one execution mode to another would involve a context switch from one process that doesn't use shared pages to another that does. P. 92 (processes)

Silberschatz et al. (p 270-271) discloses that there is an advantage to sharing common code, particularly in the context of a time-sharing environment, and that reentrant shared code can result in a significant savings of total memory space. P. 271 (paragraph 2)

Neither Silberschatz et al. or Coulouris et al. explicitly recites the limitation

- Restricting access to an isolated area of memory to bus cycles performed in the isolated execution mode.

However Silberschatz et al. or Coulouris et al. do disclose restricting access to an isolated area of memory.

A bus, is merely the path that connects the various components of a computer to allow data to be transferred from one internal component to another. All Buses transfer data in cycles as a synchronous device.

Summers et al. discloses

- Restricting access to an isolated area of memory to bus cycles performed in the isolated execution mode. (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54), where the access to the memory via the bus are also restricted with a secure bus mechanism.

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the shared code processes of Silberchatz et al. with the isolated execution processes of Coulouris et al. in order to allow for significant savings in memory while still retaining the logical boundaries of the process to allow for managed concurrent execution and to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 2:

(Coulouris et al. Section 6.3 Processes and Threads) discloses the method of claim 1 further comprising:

- Identifying if the event is one of a class of events to be handled in the isolated execution mode, where the isolated execution mode is a processor running a secure process (Page

168), and the event is one of an event or events that might be handled by that process, where threads within a process have their own software interrupt handling mechanisms

- Handling the event using the first page table map if the event is identified as one of the class of events to be handled in the isolated execution mode, where the first page table map is the virtual memory map which maps the memory for the running processes(page 169, 190-192), and the event identified as one of the events to be handled by the isolated execution mode is an event that is to be handled by that process. (page 172)
- Wherein identifying comprises indexing into a lookup table with a exception vector of the event, where the identifying of the interrupt comprises indexing the disclosed lookup table Silberschatz et al. page (404) with the interrupt or "exception" vector page (403) & Silberschatz et al. page (402-404)

In reference to claim 3:

Coulouris et al. and Silberschatz et al. discloses the method of claim 1 wherein dynamically swapping comprises:

- Loading a set of control registers selected based on an exception vector of the event, where a set control registers may be found with the data loaded from the interrupt descriptor table registers in the case of an event, where the control registers are the memory addresses of specialized interrupt handlers which are controlled by the event (exception) table. Silberschatz et al. page (402-404)

In reference to claim 4:

Coulouris et al. and Silberschatz et al. fail to explicitly disclose the method of claim 3 wherein the set of control registers comprises:

- A global descriptor table register
- An interrupt descriptor table register
- A page table map base address register.

The examiner takes as admitted prior art that a global descriptor table register and an interrupt descriptor table register were well known in the art at the time of the invention. In particular a GDTR and an IDTR are registers that contain entries which associate each interrupt or exception identifier with a descriptor for the set of instructions that are to service the event.
Both of these registers are disclosed in a number of processors and processor programming manuals include the well known 80386 Programmer Reference Manual.

It would have been obvious to one of ordinary skill in the art at the time of invention to have a GDT register and an IDT register, so that processor knows which set of instructions to use to respond to a particular event.

In reference to claim 5:
Coulouris et al. and Silberschatz et al. discloses the method of claim 1 wherein maintaining comprises:

- Mirroring a page table base address register.
- Mirroring a memory map is not explicitly disclosed however,

Silberschatz et al.(page 445) discloses a RAID organization called mirroring in which the whole disk is duplicated. While costly, the advantages of this allow reading that is twice as fast. Silberschatz et al(p. 289) also discloses that memory maps, page tables, and processes may be placed on the actual hard disk itself in virtual memory. Silberschatz et al. discloses on p. 293, Figure 9.3 that page tables and memory maps for the memory may be stored in the actual hard disk.

The mirroring a hard disk containing virtual memory on it as disclosed by Silberschatz et al. inherently discloses

- Mirroring a page table base address register.
- Mirroring a memory map is not explicitly disclosed however,

In reference to claim 6:

(Coulouris et al. Section 6.4 Naming and Protection) discloses the method of claim 1 further comprising:

- Defining a set of events that should be handled in isolated execution mode, where the set of events that should be handled by the isolated execution mode are the set of events that should be handled by a particular running process, selected by the server.

In reference to claim 7:

(Coulouris et al. Section 10.4 Distributed Coordinarion) discloses the method of claim 6 wherein the set of events to be handled in the isolated execution mode comprises:

machine check events and clock events, where the machine and clock events involve the synchronization of system clocks in a distributed system.

In reference to claim 8:

Coulouris et al. discloses the method of claim 2 wherein handling comprises:

- Determining if a current mode is the isolated execution mode, where the current mode is determined if it is in isolated execution mode, if it is determined that an isolated process is currently running. (Section 6.4 Naming and Protection)
- Loading a set of control registers with values corresponding to the first page table map if the current mode is not the isolated execution mode and the event is one of the class, where the set of control registers are loaded which contain the descriptor for the set of instructions needed to handle the current event, if it is found that the event is not to be handled by the current running process, but by another process. (Section 6.4 Naming and Protection)
- Dispatching an exception vector after the loading is complete, where the exception vector for the event is be dispatched once the new process capable of handling the event is loaded or switched to. (Section 6.4 Naming and Protection) & Figure 6.12

Claims 9-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Takahashi, US Patent 5615263 in view of Summers et al. , US patent 6098133.

In reference to claim 9:

Takahashi discloses an apparatus comprising:

- A first storage location storing control data for a first page table map, where the first page table map is the map that designates the memory. (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40)
- A second storage location storing control data for a second page table map, where the second storage location is the ROM. (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40)
- A selection unit to select which page table map is applied responsive to receipt of an event, where the selection unit chooses to select between the ROM and the memory based on the execution mode of the processor. (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40) & (Column 2, lines 45 – 61)

Takahashi fails to explicitly disclose:

- An isolated execution circuit to generate isolated access bus cycles
- Wherein isolated access bus cycles are to be used if the apparatus operates in an isolated execution mode.

Summers et. al. discloses

- An isolated execution circuit to generate isolated access bus cycles, (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54), where the access to the memory via the bus are also restricted with a secure bus mechanism.
- Wherein isolated access bus cycles are to be used if the apparatus operates in an isolated execution mode (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54) & (Column 2, line 60 – Column 3, line 15), where the access to the memory via the bus are also restricted with a secure bus mechanism.

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 10:

Takahashi and Summers et al. discloses the apparatus of claim 9 wherein the selection unit comprises:

- A multiplexer that selects between the first and second storage locations based on an exception vector of the event. (Figure 1, Item 13) & (Column 2, line 62 – Column 3, line 15)

In reference to claim 11:

Takahashi and Summers et al. (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40) & (Column 2, lines 45 – 61) discloses the apparatus of claim 9 wherein the first storage location contains a base address for the first page table map and the second storage location contains a base address for the second page table map.

In reference to claim 12:

Takahashi discloses a platform comprising:

- A processor executing in one of a normal execution mode and isolated execution mode; (Column 2, lines 45 – 61)
- A first set of control registers to define a current memory map of the platform; (Column 3, lines 45-60)
- A mapping unit to dynamically load the first set of control registers responsive to an event if the event should be handled using an alternative memory map; (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40)

Takahashi fails to explicitly disclose

- An isolated execution circuit to generate isolated access bus cycles if the processor is executing in the isolated execution mode.

Summers et al. discloses

- An isolated execution circuit to generate isolated access bus cycles if the processor is executing in the isolated execution mode. (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54),

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 13:

Takahashi and Summers et al. discloses the platform of claim 12 wherein the mapping unit comprises:

- A second set of registers having a first subset corresponding to control register values for a normal execution mode memory map and a second subset corresponding to control register values for an isolated execution mode memory map, where the isolated memory map is the ROM, read only memory containing the secure functions. (Column 3, lines 60 Column 4, lines 60)
- A selection unit to select between the first subset and the second subset. (Column 3, lines 25-47)

In reference to claim 14:

Takahashi and Summers et al. discloses the platform of claim 13 wherein the selection unit comprises:

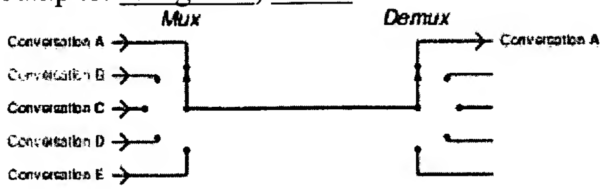
- A multiplexer having selection driven by an exception vector of an incoming event. (Figure 1, Item 13) & (Column 2, line 62 – Column 3, line 15)

However the use of multiple multiplexers is not explicitly disclosed.

The Examiner takes official notice that using a plurality of multiplexers as opposed to a single multiplexer was well known in the art at the time of invention.

Multiplexer

From Wikipedia, the free encyclopedia
(Redirected from Multiplexor)
Jump to: navigation, search



The basic function of a multiplexer: combining multiple inputs into a single data stream. On the receiving side, a demultiplexer splits the single data stream into the original multiple signals.

A **multiplexer** (or **mux** or, more rarely, **muldex**) is an encoder that combines two or more inputs into a single output. In electronics, the multiplexer combines several electrical signals into a single signal. There are different types of multiplexers for analog and digital circuits.

In digital signal processing, the multiplexer takes several separate digital data streams and combines them together into one data stream of a higher data rate. This allows multiple data streams to be carried from one place to another over one physical link, which saves cost.

In fact, multiple multiplexers may be used without any change to the input and output of a digital system as opposed to a single multiplexer if arranged to be logically equivalent. It would have been obvious to one of ordinary skill in the art at the time of invention to use multiple multiplexers to combine different size data streams into a single larger data stream.

In reference to claim 15:

Takahashi and Summers et al. fails to explicitly disclose the platform of claim 12 wherein the first set of control registers comprises:

- A global descriptor table register;
- An interrupt description table register;
- A page table map base address register.

The examiner takes as admitted prior art that a global descriptor table register and an interrupt descriptor table register were well known in the art at the time of the invention as part of a processor. In particular a GDTR and an IDTR are registers that contain entries which associate each interrupt or exception identifier with a descriptor for the set of instructions that are to service the event.

Both of these registers are disclosed in a number of processors and processor programming manuals include the well known 80386 Programmer Reference Manual.

It would have been obvious to one of ordinary skill in the art at the time of invention to have a GDT register and an IDT register, so that processor knows which set of instructions to use to respond to a particular event.

Claims 9-15 are further rejected under 35 U.S.C. 103(a) as being unpatentable over Poisner, US Patent 5729760 in view of Summers et al. , US patent 6098133.

In reference to claim 9:

Poisner discloses an apparatus comprising:

- A first storage location storing control data for a first page table map, where the first table map is the unrestricted memory as indicated by the IO mapped register. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A second storage location storing control data for a second page table map, where the second table map is the restricted memory as indicated by the IO mapped register. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A selection unit to select which page table map is applied responsive to receipt of an event, where the selection unit makes the determination based on the mode of processor execution. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)

Poisner fails to explicitly disclose:

- An isolated execution circuit to generate isolated access bus cycles
- Wherein isolated access bus cycles are to be used if the apparatus operates in an isolated execution mode.

Summers et. al. discloses

- An isolated execution circuit to generate isolated access bus cycles, (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54), where the access to the memory via the bus are also restricted with a secure bus mechanism.
- Wherein isolated access bus cycles are to be used if the apparatus operates in an isolated execution mode (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54) &

(Column 2, line 60 – Column 3, line 15), where the access to the memory via the bus are also restricted with a secure bus mechanism.

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 10:

Poisner (Column 8, line 42 – Column 9, line 15) discloses the apparatus of claim 9 wherein the selection unit comprises:

- A multiplexer that selects between the first and second storage locations based on an exception vector of the event.

In reference to claim 11:

Poisner (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67) discloses the apparatus of claim 9 wherein the first storage location contains a base address for the first page table map and the second storage location contains a base address for the second page table map.

In reference to claim 12:

Poisner discloses a platform comprising:

- A processor executing in one of a normal execution mode and isolated execution mode, where the normal mode of execution is the mode of execution that uses the unrestricted IO map and the isolated mode of execution uses the restricted IO map. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A first set of control registers to define a current memory map of the platform, where the IO memory maps are defined in the control registers. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A mapping unit to dynamically load the first set of control registers responsive to an event if the event should be handled using an alternative memory map, where the memory map are the different memories accessed depending on the different modes of execution for the processor, and each memory map is dynamically loaded based on the mode of the processor. (Figure 8) & (Figure 9) & (Figure 10) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)

Poisner does not explicitly disclose:

- An isolated execution circuit to generate isolated access bus cycles if the processor is executing in the isolated execution mode.

Summers et al. discloses

- An isolated execution circuit to generate isolated access bus cycles if the processor is executing in the isolated execution mode. (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54),

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 13:

Poisner discloses the platform of claim 12 wherein the mapping unit comprises:

- A second set of registers having a first subset corresponding to control register values for a normal execution mode memory map and a second subset corresponding to control register values for an isolated execution mode memory map; (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A selection unit to select between the first subset and the second subset. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)

In reference to claim 14:

Poisner (Column 8, line 42 – Column 9, line 15) discloses the platform of claim 13 wherein the selection unit comprises:

- A multiplexer having selection driven by an exception vector of an incoming event.

However the use of multiple multiplexers is not explicitly disclosed.

The Examiner takes official notice that using a plurality of multiplexers as opposed to a single multiplexer was well known in the art at the time of invention.

Multiple multiplexers may be used without any change to the input and output of a digital system as opposed to a single multiplexer if arranged to be logically equivalent. It would have been obvious to one of ordinary skill in the art at the time of invention to use multiple multiplexers to combine different size data streams into a single larger data stream.

In reference to claim 15:

Poisner fails to explicitly disclose the platform of claim 12 wherein the first set of control registers comprising:

- A global descriptor table register;
- An interrupt description table register;
- A page table map base address register.

The examiner takes as admitted prior art that a global descriptor table register and an interrupt descriptor table register were well known in the art at the time of the invention as part of a processor. In particular a GDTR and an IDTR are registers that contain entries which associate each interrupt or exception identifier with a descriptor for the set of instructions that are to service the event.

Both of these registers are disclosed in a number of processors and processor programming manuals include the well known 80386 Programmer Reference Manual.

- It would have been obvious to one of ordinary skill in the art at the time of invention to have a GDT register and an IDT register, so that processor knows which set of instructions to use to respond to a particular event.

Conclusion

4. The following prior art not relied upon is made of record:

- US patent 6618809, discloses a method and security system for processing a security critical activity comprising a normal and secure mode of processing.

5. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of the final action and the advisory action is not mailed under after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension pursuant to 37 CFR 1.136(A) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

6. Any inquiry concerning this communication from the examiner should be directed to Thomas M Ho whose telephone number is (571)272-3835. The examiner can normally be reached on M-F from 9:30 AM - 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gilberto Barron can be reached on **(571)272-3799.**

The Examiner may also be reached through email through Thomas.Ho6@uspto.gov

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571)272-2100.

General Information/Receptionist	Telephone: 571-272-2100	Fax: 571-273-8300
Customer Service Representative	Telephone: 571-272-2100	Fax: 571-273-8300

TMH

August 21st, 2006

GILBERTO BARRON JR
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100